SQPC / NSQ / NUSQ
DATE AND TIME ENCODING IN SQUEEZED FILES

February 23, 1985

Paul J. Homchick
Chimitt Gilman Homchick, Inc.
One Radnor Station, Suite 300
Radnor, PA 19087

I. Standardized Time and Date encoding in Greenlaw-type squeezed files.

A few Public Domain contributions have been truly remarkable, and because they have either done something first or best, have gained almost universal usage and acclaim. Ward Christensen's XMODEM communications protocol is one such contribution, Gary Novosielski's LU is another such contribution.

The third one that comes to mind is Richard Greenlaw's implementation of Huffman code compression and expansion in his SQ and USQ BDS C programs. This compression method has spread throughout the world, and is now available on CP/M, MSDOS, and UNIX systems. Any files squeezed using Greenlaw's format on any of these ASCII-based systems can be unsqueezed and recovered on any of the other systems having Greenlaw-standard programs.

When SQ and USQ were implemented on MSDOS, it became apparent that greater utility would be available if SQ and USQ supported the MSDOS time and date stamping. Since SQ and USQ predated MSDOS and were developed on CP/M (no time and date stamping), this was not available. So someone developed a new program (known as SQ2, or ZSQ) that did incorporate time and date stamping, and it began to see use on IBMPC BBS systems.

Although SQ2 added time and date stamping, it did so at the expense of downwards compatibility. A file squeezed with the time and date mode of SQ2 could ONLY be unsqueezed with the companion unsqueezer USQ2 (or ZUSQ). Thus the advantage of standardization was lost. No file squeezed with SQ2 could be unsqueezed with the older standard programs or moved to CP/M or UNIX systems. Clearly, SQ2 created a number of unfortunate consequences along with its time and date stamping.

The authors of NUSQ (an MSDOS assembly language Greenlaw-standard unsqueeze program) noted both the time and date stamping of SQ2 and its unfortunate break with the standard. They developed and implemented a method of encoding and decoding time and date stamping in squeezed files that retains complete downwards compatibility with earlier programs.

This date and time encoding was implemented in NSQ.EXE v 2.03, a compatible squeezer, and has been adopted by SQPC.COM the first assembly language squeeze program available for MSDOS. Any file squeezed by NSQ or SQPC will have its original time and date encoded in it, and yet can still be successfully unsqueezed by any standard unsqueeze program. If that file is unsqueezed by NUSQ or any compatible program, it will have its original time and date intact after the emerging from the process.

Standardization across a wide range of systems and programs is clearly advantageous, as is time and date encoding. Since the NSQ/SQPC/NUSQ method achieves both of these goals, this note is being provided to document the method used and to promote its use.

II. Specifications of the standard.

To encode the date and time of the original file in the squeezed file, add the following 4 words after the SPEOF (Special End Of File Mark).

```
    Date and Time Signature Word =        FF77
  + Date (MSDOS FCB format)      =        ????
  + Time (MSDOS FCB format)      =        ????
  = Mod 64K checksum             =        ????
```

The storage format for the date conforms to MS-DOS standards, wherein the 16-bit word comprises three sub-fields as follows:

```
 byte:  <==2==> <==1==>
 bit:   76543210 76543210
 field: yyyyyyym mmmddddd
 y = Year.  Treated as a 7-bit binary integer in the range 0..119 (1980-2099)
 m = Month.  Treated as a 4-bit binary integer in the range 1..12
 d = Day.  Treated as a 5-bit binary integer in the range 1..31.
```

The storage format for the time also conforms to MS-DOS standards, where:

```
 byte:  <==2==> <==1==>
 bit:   76543210 76543210
 field: hhhhhmmm mmmsssss
 h = Hours.  Treated as a 5-bit binary integer in the range 0..23
 m = Minutes.  Treated as a 6-bit binary integer in the range 0..59
 s = Seconds/2.  Treated as a 5-bit binary integer in the range 0..29,
     allowing resolution to the nearest 2-second interval.
```

So, 13:11:31, Feb 17, 1985 would be encoded as: 0FF77, 0A51, 696F, 7337, or as actually stored in the file in Intel format: 77 FF 51 0A 6F 69 37 73.

III. C Code from NSQ.C to install time and date at end of squeezed file.

```
/* structure for MS DOS filestamping */

struct filestamp {
      unsigned int date;
      unsigned int time;
};

squeeze(infile, outfile)
char *infile, *outfile;
{
      register INT i, c;
      FILE *inbuff, *outbuff; /* file buffers */
      INT file;
      UNSIGNED check2;

/*                                  */
/*    chunk of code removed from here          */
```

```
/*                                */

      while((c = gethuff(inbuff)) != EOF)
            if( (putc(c, outbuff) == ERROR) && (ferror(outbuff)) ) {
                  fprintf(stderr, "nsq: write error\n");
                  goto closeall;
            }
      fprintf(stderr, "done.");

      /* find out filestamp info for input file */

      if (filedate(fileno(inbuff),0,fdatePtr) < 0) {
            fprintf(stderr,"nsq: Error getting filestamp.\n");
      }

closeall:
      fclose(inbuff);
closeout:               /* install time and date at file end       */
      putwe(0xFF77,outbuff);              /* special date signature */
      putwe(fdatePtr->date,outbuff);            /* MS DOS format date     */
      putwe(fdatePtr->time,outbuff);            /* MS DOS format time     */
      check2= 0xFF77 + fdatePtr->date + fdatePtr->time;
      putwe(check2,outbuff);              /* checksum of time & date */

      fflush(outbuff);
      fclose(outbuff);

      /* stamp input file time and date on output file */

      file= open(outfile,BUPDATE);        /* binary update */
      if( filedate(file,1,fdatePtr) < 0) {
            fprintf(stderr,"nsq: Error setting filestamp.\n");
      }
      close(file);
}

/*
 * get / set file date and time
 * date and time passed in ms-dos format
 *
 * returns -1 on error, 1 on OK
 *
 * this code is specific to the Computer Innovations C86
 * C compiler version 2.20
 */

#include <fileio2.h>

filedate(fd,mode,fdate)
int fd;                       /* file descriptor      */
int mode;                /* 0 = get ; 1 = set    */
struct filestamp *fdate;      /* date and time  */
{
struct regval{unsigned ax,bx,cx,dx,si,di,ds,es;};
struct regval srv;
struct bufstr *buf;

      srv.ax= 0x5700+mode;          /* set time/date mode   */
      buf=_opentab[fd];       /* get file handle      */
      srv.bx= buf->bufhand;         /*              from C  */
      if (mode) {             /* we're setting  */
            srv.dx= fdate->date;
```

```
            srv.cx= fdate->time;
      }
      if (sysint21(&srv,&srv)&1)
            return(-1);
      if (!mode) {                  /* we're getting  */
            fdate->date= srv.dx;
            fdate->time= srv.cx;
      }
            return(1);
}

/*
 * machine independent put-word that writes low order byte first
 *  (compatible with CP/M original) regardless of host cpu.
 */

putwe(w, iob)
INT w;
FILE *iob;
{
      putc(w, iob);
      putc(w>>8, iob);
      if (ferror(iob)) {
            fprintf(stderr, "nsq: write error\n");
            exit(1);
      }
}
```

IV. NUSQ date and time options from NUSQ manual.

OVERVIEW

MS-DOS versions of NUSQ including and subsequent to 1.10 can use date and time information encoded in files squeezed with NSQ or SQPC. This means that a file squeezed with NSQ, 'modemed' to another location, and unsqueezed with NUSQ (version 1.10D or greater) will retain the date and time of ORIGIN instead of DESTINATION. NSQ encodes this information at the END of the squeezed file, so such files RETAIN COMPATIBILITY with earlier squeeze and unsqueeze programs.

See the installation section of this document for details on how to implement this feature in NUSQ. Any files squeezed with NSQ (versions 2.01 or higher) or SQPC (1.28 is the first bug-free version) will have the necessary date and time information encoded in them.

INSTALLATION

Select Date Option allows you four choices for the way the output file is dated. As distributed, NUSQ will date the output file in the same manner as most other programs do, i.e. the date and time shown in directory displays will be the date and time at which you unsqueezed the file. You may change this option by changing the byte at location 0104H. Copies of NUSQ patched in this manner will display a 'D' after the version number, signifying that the input file date option has been enabled, i.e. "1.10D".

The available options are:

0) Leave it to DOS
1) Output file gets the same date/time as the input file
2) If the squeezed file contains the special date/time
   signature inserted by NSQ or SQPC at squeeze time,
     use the date/time contained in that signature;
     otherwise, use the current date/time
3) If the squeezed file contains the special date/time
   signature inserted by NSQ or SQPC at squeeze time,
   use the date/time contained in that signature;
   otherwise, use the same date/time as the input file

If you would like the unsqueezed file to retain the same date and time shown for the input file (option 1), patch 0104H to 0FH. For option 2, patch 0104H to 0F0H; for option 3, patch 0104H to 0FFH.

Following is a table that explains these options:

option	value	use current	use input file	use encoded
0.	00	yes	-	-
1.	0F	-	yes	-
2.	F0	-	-	yes
3.	FF	-	do this second	try this first

There is a reason for NUSQ being distributed with the date/time options disabled. If you have a backup utility (usually this applies to a "hard" disk) that uses the file date and time to determine whether or not a file should be backed up, it is possible that using the input file date could cause loss of a file by having an 'old' date on a 'new' file, and losing a backup. If you are unsure whether this applies to you, leave the date/time options unchanged until you can find out.